Density-Aware Sliders

Name
Affiliation
Address
Email
Phone

Name
Affiliation
Address
Email
Phone

Name
Affiliation
Address
Email
Phone

ABSTRACT

We present a selection widget that utilizes pre-existing knowledge of the distribution of the underlying data to ease browsing and selection. Standard sliders suffer from two problems: poor subpixel data querying, and uniform visual representation of non-uniform data. We provide density-aware sliders to give users more efficient subpixel data querying and use embedded visualizations to introduce or improve the visual representation of the data contained by the slider. Through a controlled user study, we find that our proposed density-aware slider outperforms the standard Alphaslider with large datasets.

Author Keywords

Dynamic query, information visualization, slider, data selection, Alphaslider, lasso, range slider, density-aware

ACM Classification Keywords

H5.2. Information Interfaces and presentation (e.g., HCI): User Interfaces.

INTRODUCTION

Dynamic queries provide easy to use, powerful and efficient tools and interfaces that allow users to rapidly and reversibly query data and uncover trends in the data being explored [1]. Currently, there are many tools available to users that benefit from the advantages of dynamic queries such as the slider, checkbox, or button. The Alphaslider is a slider designed to query large lists of alphanumerically sorted alphanumeric items [2].

The sliders in general suffer from two problems.

1) They do not have an awareness of the data they contain.
2) They give users poor visual feedback

These problems are especially important when the interface maps more than one item to any pixel along the slider track. Subpixel querying is a new problem because information density is ever increasing. Data selection is becoming increasingly difficult because each pixel maps to more items.

The PVSlider [3] and FineSlider [4] both aim to improve upon the Alphaslider. The PVSlider uses a popup vernier to give users sub-pixel-pitch control and sub-pixel visualization. The FineSlider uses an elastic band metaphor to give users more intuitive control over scrolling speed and precision where a longer elastic band correlates to faster scrolling and a shorter band correlates to more precise and deliberate item selection. The Zlider, proposed by Ramos et al, takes advantage of pressure input in a pressure sensitive environment to fluidly change from coarse to fine granularity and shift granularity control from the system to the user [5]. None of the above sliders dynamically adapt to the data they contain. We believe that the subpixel visual feedback provided by the PVSlider is inadequate because the slider does not dynamically adapt itself.

We propose density-aware sliders that are knowledgeable of the data they contain and use this information to simplify data querying. Our sliders intelligently redistribute the items in densely packed pixels over a larger area and provide users with a visualization that gives them a sense of location in the pixel, and provide a better querying and selection method. To solve the issue of data visualization we used embedded visualizations such as those proposed by Willett et al [6]. These visualizations aid in estimation of

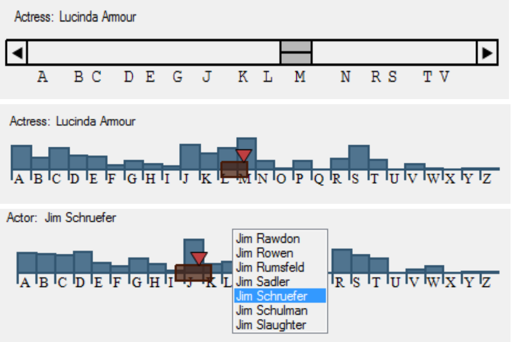

Figure 1: (Top) Display Distortion with the Alphaslider. Letter spacing indicates distribution and users have constant input. (Middle) Input Distortion with the ActiveArea Slider. Histograms indicate distribution and user input gets distorted based on local density. (Bottom) The ActiveList Slider with an Input Distortion

Submitted to CHI 2013 – this had the copyright info previously. Should I make this textbox smaller? Or keep it the same size for when the copyright info is put back in?

the density of items in a given pixel.

The results of our quantitative study show that participants can query 3.4 seconds (or 15.5%) faster with our density-aware sliders compared to the Alphaslider on a dataset of 60 000 elements. Our results are statistically significant (Fill in this statistics data).

DESIGNING OF MULTI-ITEM PIXEL ASSIGNMENT

Mapping multiple items to a single pixel and giving users quick access to each item remains an ever-present problem. The Alphaslider gets around this issue by giving users list based movement where the user navigates the list of items contained by the slider through coarse or fine-grained dragging. Coarse dragging lets the user skip through the list 10 items at a time while fine-grained dragging lets the user move through the list 1 item at a time. This causes a problem to arise when multiple lists are present of different densities. A slider containing 10 000 items will feel heavier than a list containing 1 000 items causing difficulty for the user in predicting how quickly they need to drag the slider. The Alphaslider's querying method gives no visual feedback when the items per pixel exceeds the coarse grained movement value making highly dense data even more confusing.

Subpixel visual feedback is also an issue intricately associated with the Alphaslider. This problem was identified by Ayatsuka et al [3] and they attempted to rectify it. However, their solution lacks the ability to adapt dynamically to the contained data. The popup vernier in the PVSlider requires that the developer be explicitly aware of the information to give the slider appropriate scales for the popup vernier.

Approaching this problem with multi-level querying in mind can solve the above issues. The issue of slider weight is solved by providing the user with multi-level querying because the slider's movement is pixel based instead of item based. Items per pixel becomes a non-issue in terms of visual feedback because coarse querying is pixel based instead of item based. Subpixel querying is partially solved because items mapped in a single pixel are instead mapped along the width of slider which can be individually selected. We believe that adding a list which enumerates items being mapped along the slider further refines our solution.

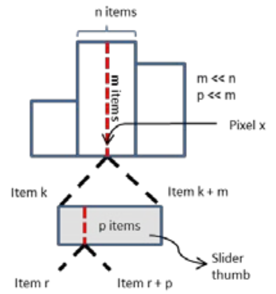

Figure 2: Visual representation of our multilevel approach to subpixel querying. Pixel x contains m items which are mapped across the slider thumb. Each horizontal pixel of the thumb contains p items.

Comment [p1]: Rephrase this paragraph.

EXPERIMENTAL DESIGN

We conducted an experiment to compare different designs of density-aware sliders against the Alphaslider.

Apparatus

We built the interfaces for the experiment using Visual Studio. Participants used a standard 3-button mouse and a 23.5" Dell monitor with a resolution of 1920x1080 pixels. The experimental system consisted of an Intel i5-2400 CPU with a clock speed of 3.1 GHz and 4 GB of RAM. Twelve participants completed four trials for each condition. Participants ranged from 20 to 39 years of age.

Comment [p2]: I had to change a lot of passive voice to active voice. Input on my phrasing would be nice.

Interfaces

The experiment used three different sliders and two distortion styles (Figure 1).

Alphaslider

The Alphaslider (Figure 1 (top)) has four navigation techniques. Users can jump directly to an item in the list by clicking anywhere in the bounded area above the letters. Users can navigate through the list at a rate of ten items per mouse movement by clicking in the top tile of the slider thumb and dragging. Users can also navigate one item at a time either by clicking on the arrows at the ends of the slider or by clicking in the bottom tile of the slider thumb and dragging.

ActiveArea Slider

The ActiveArea Slider (Figure 1 (middle)) allows users to navigate by clicking on and dragging the slider thumb, by clicking on and dragging the secondary red slider or by pressing the left or right arrow keys on the keyboard. The main slider has a variable size which changes based on the density of information of the pixel it queries mapping a maximum of four items for each horizontal pixel of the slider. Users can drag the red triangle (a secondary slider) to query items mapped to the main slider. Users can also roll the mouse wheel to navigate one item at a time. Rolling the mouse wheel appropriately moves the secondary slider and gives users continuous querying whereas the edges of the main slider bind the dragging of the secondary slider.

ActiveList Slider

The ActiveList Slider (Figure 1 (bottom)) incorporates a list into the ActiveArea Slider. The queried subpixel item determines the contents of the list. The items of the list are the items r through r + p. If p is less than a minimum threshold then the list is enumerated up until that threshold is met or until item k + m is shown. Dragging the red slider and rolling the mouse wheel appropriately update the list. The edges of the main slider provide physical limits to querying by dragging the secondary slider while the mousewheel lets users perform continuous querying.

Input Distortion

This distortion (Figures 1 (middle and bottom)) uses histograms to give users information about item distribution. We call this Input Distortion because users have to distort their input based on the local density.

Display Distortion

This distortion (Figure 1 (top)) uses variable spacing between letters to give users information about item distribution. Users do not have to distort their input because item density is uniform.

Hypotheses

This paper is primarily concerned with designing the most efficient slider. While accuracy is an important factor in designing a slider, that attribute falls largely upon the user. Because of this speed is the most significant factor to measure. The speed with which a user locates an item is largely dependent on total mouse movement. For each of the sliders there is a period of querying where the user approximates the area of the target and a period of querying where the user does fine adjustments to acquire the target. Based on this assumption we make the following hypotheses:

H1) The ActiveList slider will perform the best because it does the best job breaking the query into multiple levels.

H2) All of our sliders will outperform the Alphaslider because they give the user better control and visual feedback.

H3) The ActiveList Slider will be preferred because the technique gives users better, quicker and easier access to the data.

Experiment Design

Independent Variables

1. Technique (*Alphaslider (AS)*, *ActiveArea Slider (AAS)* or *ActiveList Slider (ALS)*)
2. Distortion Style (*Input Distortion (ID)* or *Display Distortion (DD)*)
3. Local Density (*Low* or *High*)

Dependent Variables

1. Target acquisition time
2. Error rate
3. Perceived difficulty of use
4. Subjective satisfaction

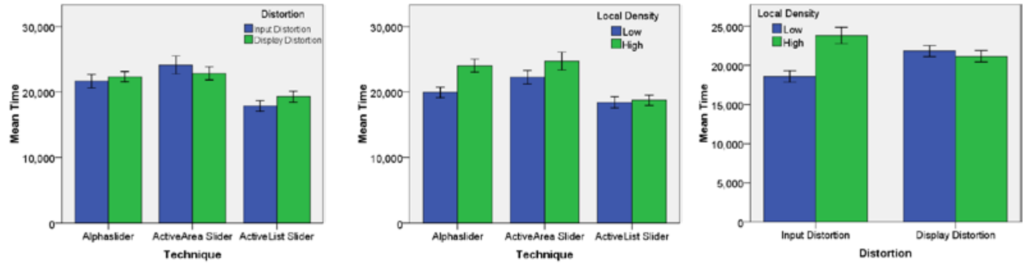

Figure 3: (a) Graph of average Time vs. Technique clustered on Distortion Style. (b) Graph of average Time vs. Technique clustered on Local Density (c) Graph of average Time vs. Distortion clustered on Local Density

Tasks

The system provided users with four randomly generated search targets - one at a time - for each condition. The search target would be a name that the users had to find using the appropriate slider. Upon completion the task users would press the spacebar and be given a new task. With each new task, the slider thumb returned to the beginning of the track.

Comment [p3]: Should this be rephrased?

Procedures

We first ran a pilot study to filter poorly designed and inefficient interfaces. Prior to the timed trials the participants were given a set of tasks under each condition (the ordering of which was determined by a balanced Latin square) to familiarize themselves with each interface while reading interface specific instructions and ask questions. During the timed trial, the experimenters disallowed participants from asking questions and required them to fill out a NASA TLX form for each technique and distortion style. Upon completion of the experiment, the participants filled out forced pairwise comparison sheet on which they indicated their preferred interface for every combination of technique and distortion style. The experimenters also asked participants to write down any comments they have regarding the experiment, techniques, and distortion styles.

RESULTS

We used a univariate general linear model for our ANOVA analysis of acquisition time and error rate. We performed a Bonferroni post-hoc pairwise comparison (equal (?) variances) on the acquisition time data. We used Friedman's X^2 test to analyze the TLX data.

Acquisition Time: Univariate ANOVA reveals significant main effect of technique ($F_{(2, 22)} = 7.548$, $p = 0.003$) and of local density ($F_{(1, 11)} = 5.432$, $p = 0.04$). The ANOVA analysis also reveals significant interaction effects between local density and distortion style ($F_{(1, 11)} = 16.672$, $p = 0.02$). We believe this interaction effect arises because of the fact that the display distortion equalizes the density between indices. Post-hoc pairwise analysis showed significance between ALS and AS ($p < 0.001$) and between ALS and AAS ($p = 0.001$).

Error Rate: We found no significant difference in error rate among techniques ($F_{(2, 22)} = 1.000$, $p = 0.384$) or among distortion styles ($F_{(1, 11)} = 0.000$, $p = 1.000$). Participants were 98% accurate with the Alphaslider and 99% accurate with the ActiveArea Slider and the ActiveList Slider. Participants were also 99% with both distortion styles and both densities.

TLX Analysis: Analysis of the TLX information indicates that there were significant main effects for frustration and effort among interfaces (X^2 (2) = 7.386, $p < 0.05$ and X^2 (1) = 18.286, $p < 0.001$, respectively). No significant main effects were found for distortion style for either frustration or effort (X^2 (2) = 0.034, $p = 0.853$ and X^2 (1) = 0.533, $p = 0.465$, respectively). Significant interaction effects were found for effort (X^2 (5) = 22.134, $p < 0.001$) and for frustration (X^2 (5) = 12.481, $p < 0.03$). Post-hoc analysis with Wilcoxon Signed-Rank Tests and Bonferroni corrections revealed statistically significant differences in effort between the ActiveList Slider with both distortions and every other condition ($p < 0.01$). The post-hoc analysis also revealed no significant differences for frustration. This information supports H3.

DISCUSSION

Our results show statistically significant differences between the techniques. Specifically, a difference is found between the ActiveList Slider and Alphaslider and the ActiveList Slider and ActiveArea Slider ($p <= 0.001$). Our results do not show significant differences between distortion styles.

Overall, participants preferred our sliders to the Alphaslider. This is because the Alphaslider requires that users have fine motor skills to query one item at a time or to click on arrow buttons repeatedly while our interfaces let users use the mousewheel, which is highly precise and requires less motor skills without sacrificing speed. Many users commented, “I really like using the mousewheel” and even sometimes instinctively attempted using the mousewheel while querying with the Alphaslider. One user sighed, “Oh good, the one with the list” in relief during the experiment.

Lessons Learned

We learned that while users prefer using the input distortion style they performed marginally but insignificantly better with it. We believe that the added complexity of requiring variable input offsets the visual advantage it provides over the display distortion. We also learned that users heavily prefer using a mousewheel for fine-grained querying to cursor movement. We believe that this is because of the higher motor precision required to move the cursor.

Limitations

We suspect that with ultra-dense datasets (greater than 100 000 items) we will start seeing performance degradation with our sliders because of the instinctive behavior of users. Ideally, users should use the different granularities provided by our sliders: the main thumb, the secondary thumb, and the mousewheel. Users tend not to use the secondary thumb because they perceive it as having the same or similar granularity as the mousewheel, which is true for small to medium density datasets but not for larger datasets. At approximately 20 000 items the granularity difference between the mousewheel and secondary thumb starts becoming noticeable. We believe that explicit training or a new visual cue can solve this problem.

Applications

One major application of our work can be in video editing where editors need to select from tens or hundreds of thousands of frames to edit. Our sliders can give editors highly precise frame selection while still adapting to movies of varying length or movies with variable frame rates. Another possible application lies in the world of math with precise visual integration. Our sliders can allow users to set upper and lower bounds for visual integration while also giving users the flexibility to adjust the precision on either end of the integral independently while receiving appropriate visual feedback.

CONCLUSIONS AND FUTURE WORK

We presented density-aware sliders and a new slider distortion style that were meant to reduce querying time while keeping user error rates constant compared to the Alphaslider. We found that our new distortion style was ineffective but that our proposed density-aware sliders show promise for large datasets. We plan explore under what conditions our density-aware sliders excel. In the future, we would also like to investigate other possibilities for density-aware widgets such as a range slider or lasso selection tool.

AKNOWLEDGEMENTS

We thank NSERC for funding part of this research and the participants for the valuable time and effort.

REFERENCES

[1] Shneiderman B., "Dynamic Queries for Visual Information Seeking," *IEEE Software*, vol. 11, no. 6, pp. 70-77, November 1994.

[2] Christopher Ahlberg and Ben Shneiderman, "The Alphaslider: A Compact and Rapid Selector," in *Human Factors in Computing Systems*, Boston, 1994, pp. 365-371.

[3] Yuji Ayatsuka, Jun Rekimoto, and Satoshi Matsuoka, "Popup Vernier: a Tool for Sub-pixel-pitch Dragging with Smooth Mode Transition," in *Proceedings of the ACM Conference on Human Factors in Computing Systems*, 1998, pp. 39-48.

[4] Toshiyuki Masui, Kouichi Kashiwagi, and George R. IV Borden, "Elastic Graphical Interfaces to Precise Data Manipulation," in *CHI Conference Companion on Human Factors in Computing Systems*, 1995, pp. 143-144.

Comment [P4]: I know that I’m missing the second degree of freedom. I’m currently working from home and I know that another person in the lab (khalad) is currently using SPSS on the remote machine meaning I can’t login into that machine to access the data again. I will fix this tomorrow morning, September 12 2012.

Comment [P5]: Should I include F?

Comment [P6]: Not entirely sure what to put here.

[5] Gonzalo Ramos and Ravin Balakrishnan, "Zliding: Fluid Zooming and Sliding for High Precision Parameter Manipulation," in *Proceedings of UIST*, 2005, pp. 143-152.

[6] Wesley Willett, Jeffrey Heer, and Maneesh Agrawala, "Scented Widgets: Improving Navigation Cues with Embedded Visualizations," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1129-1136, November 2007.

The columns on the last page should be of approximately equal length.
Remove these two lines from your final version.